Computing Machinery I
Assignment 5
8% of your final score
Due December 5th @ 11:59PM MST

Lead TA
The lead TA for this assignment is Kanishka Singh (Kanishka.singh1@ucalgary.ca)

Objective
The objective of this assignment is to write modular code in ARMv8 assembly, utilizing subroutines and value and reference parameters.

New Skills needed for this assignment

- Ability to use subroutines
- Ability to use parameters (passed by value and by reference)
- Ability to use command line arguments in assembly
- Ability to write and read files in assembly

Overview
Re-write Assignment 1 in ARMv8 assembly. For your convenience, we will reproduce the assignment details below.

Details
Display to the user an M×N table of random/given integers between 0 and 9(inclusive). The user provides M and N integers between 5 and 20 and optionally the file name containing an existing table, if applicable, using command line arguments, such as (if occurrences are randomly generated):

```
./assign5  5  6
0  1  0  2  3  1
2  5  1  0  0  0
0  7  8  0  2  7
0  0  0  5  4  0
5  8  0  0  0  9
```

In this example, there are 5 documents and 6 possible words. Document 1 contains 2 occurrences of word 0, 5 of word 1, and 1 of word 3, and no occurrences for words 3 to 5.

If occurrences are read from a file named *occurrences.bin*, then the name of the file is also specified:

```
./assign5  5  6    occurrences.bin
9  1  0  0  0  0
1  3  5  8  5  3
7  3  9  0  0  0
0  0  0  5  4  5
0  0  1  6  3  1
```

Occurences.bin >> This dataset file used to evaluate your assignment contains 401 int values in the range 0 and 9 (inclusive). Elements for the grid will be read based on M and N provided by the user using the command line arguments.

For example:

EG-1

```
./assign5  5  5    occurrences.bin
9  1  0  0  0
1  3  5  8  5
7  3  9  0  0
0  0  0  5  4
0  0  1  6  3
```

EG-2

```
./assign5  6  6    occurrences.bin
9  1  0  0  0  0
1  3  5  8  5  3
7  3  9  0  0  0
0  0  0  5  4  5
0  0  1  6  3  1
6  3  1  0  3  2
```

The user then enters the index of the word s/he is searching for and the number of the top documents containing the word, as follows:

```
Enter the index of the word you are searching for: 5
How many top documents you want to retrieve? 2
```

Your program will find the 2 top documents where the frequency of the searched for word (with index 5 in the example above) is the maximum among all documents.

Frequency of word j in document i is: occurrences[i,j] / size of document i

The size of document i (row i) is the sum of occurrences of all of its words (sum of all columns in row i).

Use any sorting algorithm. The user can quit the program or choose to search again. A log file must be created showing: the initial table, user input and the search result before exiting the program.

You do not need to worry about ties: they can be sorted in any order

Modularity

Your code must divided into closed subroutiunes as appropriate. At a minimum, you must define the following functions (we are not showing all necessary arguments):

- `initialize(*table)`
- `randomNum(m,n);` `m` and `n` are the lower and upper bounds for the random number. You can use the C library function `rand()`.
- `display(*table)`
- `topRelevantDocs(*table, n)`
- `logToFile()`

P.S.- You are not allowed to "write your own" C functions for the assignment and thus not allowed to link C and assembly code to complete this assignment. However, you are free to use built-in functions of C like printf, scanf, rand()...

Submission

- **Note:** The TA may provide further submission instructions

- Name your program *assign5.asm*
- Create a script file and call it *assign5.script.* **The script file must contain a GDB session.**
- Name your log file *assign5.log*
- Submit a *README* file providing extra instructions or information for your TA, such as the sorting algorithm you are using
- Submit your work to the appropriate dropbox on D2L.

Late Submission Policy

Late submissions will be penalized as follows:
-12.5% for each late day or portion of a day for the first two days
-25% for each additional day or portion of a day after the first two days
Hence, no submissions will be accepted after 5 days (including weekend days) of the announced deadline.

Academic Misconduct

This assignment is to be done by individual students: your final submission must be your own original work. Teamwork is not allowed. Any similarities between submissions will be further investigated for academic misconduct. While you are encouraged to discuss the assignment with your colleagues, this must be limited to conceptual and design decisions. Code sharing by any means is prohibited, including *looking* at someone else's paper or screen. The submission of compiler generated assembly code is absolutely prohibited. Any re-used code of excess of 5 lines in C and 10 lines in assembly (10 assembly language instructions) must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks

Marks posted on D2L are subject to change (up or down).

Computing Machinery I

Assignment 5 Rubric

Student:____

Item	Max Points	Points
Code compiles	5	
Code runs	5	
Correct results	20	
Log file functionality	10	

User interface (input validation, implementing all features)	15	
Random numbers	10	
Modularity	15	
Command-line arguments	5	
Passing parameters by reference	5	
Code readability (formatting documentation)	10	
Script file	5	
Total Points	**100**	